Routing in Networks with Random Topologies

Keith Scott Nicholas Bambos
{kscott,bambos}@ee.ucla.edu

Department of Electrical Engineering
[University of California at Los Angeles
Los Angeles, CA, 90024-1594

ABSTRACT: We examine the problems of routing and server assignment in network with random connectivities. In such a network the basic topology is fixed, but during each time slot and for each of its input queues, each server (node) is either connected to or disconnected from each of its input queues with some probability. During each time slot a server must decide which of its connected input queues to serve and into which of its output queues the completed job should be placed.

For single-input single-output acyclic queueing networks with random connectivities, we show that the joint routing/service policy that routes customers along the least populated path to the destination and serves any non-empty queue maximumizes throughput. We briefly discuss some implementation aspects of the proposed policy, including its robustness in stabilizing the system with respect to delayed state information.

1 Introduction

Many types of communications systems exhibit time-varying connectivity among the various nodes. These include terrestrial mobile wireless systems, meteor-burst communications systems, and environments with hard interference constraints such as a manufacturing floor. If the individual satellites of a satellite network are allowed to modify their orbits or the scheduling of a limited number of antennae for communicating with their neighbors, this would also generate connectivities that vary randomly with time.

Previous works have studied the queueing statistics for a system with a single server with randomly varying connectivity to parallel queues [1, 2]. In addition, [2] examines networks of such nodes under general assumptions about the input and service distributions of jobs/customers. In [3], the authors use coupling to examine the problem of routing to parallel finite-length queues with random connectivity to a single server to minimize the system's loss flow.

In [2] the authors compute the maximum stable input rate to an acyclic system of queues with randomly varying connectivities by use of a linear program. After solving the program, jobs are stochastically routed at the output of each server to achieve specified average flows across the various links in the network, stabilizing it. Unfortunately, the complexity of the linear program grows exponentially with network size and requires precise knowledge of the connection probabilities. Thus it may be difficult in cases where the connection probabilities are unknown and/or varying to first estimate them and then solve for the optimum flow rates with which to route customers. It is for these reasons that we seek a simple, stationary joint routing/service policy.

In this paper we show that for single-input single-output acyclic. systems with random link connectivities as illustrated in figure 1, routing along the least populated path to the destination and serving any non-empty connected queue stabilizes the system whenever possible. 'l'hat is, upon service completion, the server calculates the number of customers in each path from itself to the destination and routes the newly completed customer along the path containing the fewest jobs. This routing scheme requires that a node know all of the queue lengths of the system, but not the connection probabilities of the links. Thus it responds dynamically to changes in the network connection probabilities.

The rest of this paper is organized as follows. In section 2 we present our model for networks with random connectivities. Section 3 gives the proof that LPP routing stabilizes the system. In section -1 we briefly touch

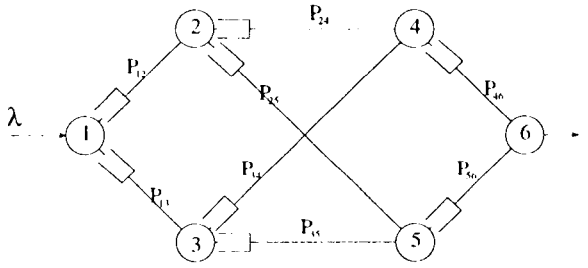

Figure 1: A network with random connectivities. here P_{ij} gives the probability that server j is connected to server i's output queue during a given time slot.

on some implementation details. Section 5 gives some concluding remarks and discusses our ongoing work.

2 System Model

We consider a single-input single-output acyclic network composed of N nodes (servers); N = {1, 2, ... N} as shown in figure 1. lime is slotted, with slots indexed by $t \in \mathcal{Z}_+$. All jobs enter the system through a single node, and we define A^t as the number of arrivals to the system up to and including time t. 'J'here is at most one arrival during any given slot.

Each server i is connected to a set **Pi** of parents and a set C_i of children by FCFS buffers. The set of all such buffers is denoted by Q and its cardinality by $|\mathbf{Q}|$. We refer to the queue between nodes i and j as either one of node i's child or output queues, or as a parent or input queue for node j. For generality we will allow multiple parallel connections between pairs of servers.

We assume that a node always has access to (is connected to) each of its output queues. For the connections between nodes and their parent queues, we define the binary connectivity variables $\{M^t_{i,j,k}\}$. If $M^t_{i,j,k} = 1$ we say that server j is connected to the k^{th} parallel queue connecting servers i and j during slot t, in which case it can serve customers from that queue. If $M^t_{i,j,k}$ is O we say that server j is not connected to the k^{th} output queue of server i during slot t. Thus the connectivities for the slot are set before the server makes its decision about which queue to serve. When there is only one queue connecting a pair of servers we drop the third index.

We can define a one-to-one correspondance between the triples (i, j,k) that appear in the definitions of the connectivity variables aud the indices of the queues in Q. Thus we can write $\mathrm{P}(M_{i,j,k} = 1) = P^c_q$ for some $q \in \{1, 2, \ldots, |\mathbf{Q}|\}$. Throughout this paper we make the following assumptions:

A *1:* The probability of an arrival during any given slot is λ. For each triplet (i, j,k), the variables $M^t_{i,j,k}$ are taken to be tile states at time t of two-state Markov chains which are independent of each other and the arrival stream. We denote by $P^c_{i,j,k}$ the probability that $M_{i,j,k} = 1$. The service times of all jobs are equal to the slot duration, so that if a server chooses to serve a given customer from a connected input queue, th$_e$ customer will complete service by the end of the time slot and, if the server is not the destination node for the network, will be placed in one of the server's output queues. The operation of choosing the output queue into which the customer will be placed is routing.

As the system state we consider the vector **Y(t) = (X(t), C(t))** where X(f) $= (X_q(t), q = 1, 2, \ldots, |\mathbf{Q}|)$ is the vector of queue lengths and C(f) = $(C_q(t), q =$ 1,2 ,..., $|\mathbf{Q}|)$ is the vector of server-queue connectivities at time t. That is, **if** $C_q(t) = 1$ we say that the server at the tail of queue q is connected to it and has the option of serving it.

We seek a stationary policy π that bases the routing and allocation decisions of each server during each slot on the information in Y(t). We denote the set of such stationary policies by G. A policy π is nothing more than a function $\pi : \mathcal{Y} \in \mathbf{Y(t)} \rightarrow \mathbf{U(t)} = (s_i(t), k_i(t), r_i(t)) : i = 1,2,\ldots, N)$ where $(s_i(t), k_i(t), r_i(t)) = 0, k, l)$ indicates that server i serves its k^{th} parent queue from node j during slot t and routes the customer to its child queue destined for node 1. If a particular server i does not serve any customer during a slot then we define $s_i(t) = k_i(t) = r_i(t) = -1$. The destination node routes customers to an imaginary node with id -2.

Remark 2.1 *Under the assumptions A 1 and using a stationary routing/service policy, the system state* $\mathbf{Y}(t)$ *is a Markov chain.*

Definition 1: We *define* the system to be stable under some routing/service policy $\pi \in$ G if the queue length process X (t) is irreducible and has a probability distribution that converges in the sense that

$$\lim_{t \rightarrow \infty} P[\mathbf{X}(t) \leq \mathbf{b}] = F(\mathbf{b}), \forall \mathbf{b} \in \chi \qquad (1)$$

where $F(\cdot)$ is a probability distribution on χ.

Definition 2 : We say that the system is *stabilizable* **if** there exists a policy in G under which it is stable.

3 Stability under LPP Routing

Theorem 3.1 *For single-input single-output acyclic networks with random connectivities, the joint routing/allocation policy that routes customers along the least populated path to the destination while always serving a non-empty input queue (if one exists) stabilizes the system whenever it is stabilizable. When two or more paths to the destination contain the same number of jobs, one is chosen at random.*

The example in figure 2 illustrates the need to route along the shortest path rather than to the shortest output. queue. Under the random connectivity assumptions, some servers may be very willing to accept new jobs while being unable to forward them. These servers could maintain very short input queues while hoarding customers in their output queues. Since they maintain short input queues, parents of these servers would continue to route customers to them, even though those customers are collecting and are not leaving the network. By routing customers along the least populated path to the destination we avoid this problem.

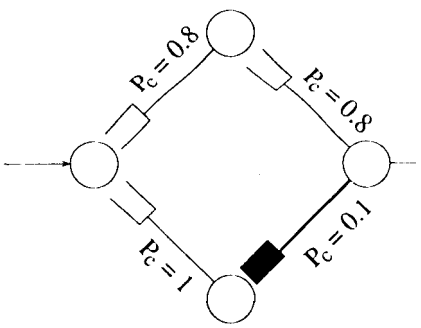

Figure 2: An example of a hoarding server. Here the lower server always tries to grab customers from its input buffer, but they get trapped in its output buffers.

Now consider a single-source, single destination, single commodity acyclic network with random connectivities between servers and their input queues. A cutset C is defined as a set of links in the network such that every path from the source to the destination traverses at least one link in C.

Each cutset C, composed of N_C links, defines a set S_C of servers composed of those servers at the tails of the links in C (i.e. those servers that serve at least onc link in C). We label these servers as S_C^i for i = 1,2, ... N_{S_C}. For a particular server S_C^i connected to $J_{S_C^i}$ links in C, let its connection probabilities to those links be given by $P(C, i, j), j$ = 1,2,.. . J_{C^i}. Note that the server S_C^i may be connected to links not in C as well.

Each cutset C has associated with it a maximal flow f_C, defined as the maximum rate at which customers can be served from the links in C. This *maximal* rate is purely a function of the connection probabilities and does not depend on the service or routing disciplines and is defined as follows:

$$f_C = \sum_{i=1}^{N_{S_C}} f_{S_C^i} \tag{2}$$

$$f_{S_C^i} = 1 - \prod_{j=1}^{J_{C^i}} (1 - P(C, i, j)) \tag{3}$$

3.1 Necessity

Lemma 3.1 *Let $f^* = \min_{C \in \mathcal{C}} f_C$ where $\mathcal{C}$ is the set of all cutsets. If $\lambda > f^*$ then the system is unstable.*

Proof : The set $\mathcal{C}$ is not empty. Assume $\lambda > f^*$ and let A^t be the number of arrivals to the system in $[0, t]$. From our assumptions, $\lim_{t\to\infty} At/t + \lambda$. Since $\lambda > f^*$, we have $\lambda > f_{C'}$ for some C'. Let S^t be the number of potential service opportunities to links in C' during [0, t], then

$$\lim_{t\to\infty} \frac{S^t}{t} \to f_{C'} < \lambda \tag{4}$$

If N^t is the number of customers in the system at time t,

$$N^t \geq A^t - S^t \tag{.5}$$

and taking the limit as $t \to \infty$ and dividing by t gives:

$$\liminf_{t\to\infty} \frac{N^t}{t} \geq \liminf_{t\to\infty} \frac{A^t}{t} - \liminf_{t\to\infty} \frac{S^t}{t} \tag{6}$$

$$= \mathbf{A\text{-}J\text{-}*>0} \tag{7}$$

■

3.2 Sufficiency

Lemma 3.2 *For $\lambda > f^*$, customers accumulate in the system at a rate less than or equal to $\lambda - f^*$. That is:*

$$\lim_{t\to\infty} (\#\ of\ jobs\ in\ system)/t \leq \lambda - f^* \qquad (8)$$

Thus for arrival rates $\lambda < f^*$ the system is retestable.

Let $\mathcal{Z}$ be the set of all paths from the source to the destination, with Z_j^t denoting the number of customers in the system at time t who are traversing path $\mathcal{Z}_j$. At each node, customers are routed along the least populated pathso that in particular a customer arriving at time t will be routed along the path $\mathcal{Z}_i$ where $Z_i^t = \min_j Z_j^t$.

Remark 3.1 *Since customers are routed independently at each node, we do not know on arrival exactly which path a customer will take. For any given customer however, we can consider running the system until that customer exits, at which time its path is determined.*

Remark 3.2 *Although Z_j may be zero, this does* not *mean that there are no customers in any of the queues comprising $\mathcal{Z}_j$, as other paths may share links with $\mathcal{Z}_j$.*

Lemma 3.3 $\max_i Z_i \to \infty \Rightarrow Z_i \to \infty\ \forall i$

Proof : Suppose that $\max_i Z_i \to \infty$ and that there were a $\mathcal{Z}_j$ such that $\liminf_t Z_j^t < \infty$. Under our assumptions, $\lim_{t\to\infty} Z_j^t$ exists, so $\limsup_t Z_j$ is also finite. Note that under LPP routing, customers are only routed along the longest path when all path lengths are equal, and consider the routing decisions of the source node.

Since $\liminf_t \max_i Z_i^t = \infty$ and $\limsup_t Z_j^t < \infty$, there must be a time T' such that for all $t \geq T'$, $\max_i Z_i^t > Z_j^t$. But then we would have $\limsup_t \max_i Z_i^t = \max_i Z_i^{T'}$, since no customers would be routed along the longer paths after time T'. This contradicts the assumption that $\max_i Z_i \to \infty$, so we must have $Z_j \to \infty$ Vj. ■

Assume $\lambda > f^*$. For any cutset C, let Y_C be the number of customers who have already crossed C. Define the most forward cutset C_f as the cutset for which 1) all of the queues defining the cutset are blowing up and 2) $\limsup Y_{C_f}^t$ is finite. There must be at least one path for which the number of customers is growing without bound by lemma 3.1, and lemma 3.3 shows that that number of jobs in every path must be increasing with time. These together with the Markovian assumptions imply the existance of C_f.

Let $N_{C_f}^t$ be the number of customers who have not yet traversed the links in C_f

$$N_{C_f}^t = A^t - \text{flow across } C_f \text{ in [0,t]} \qquad (9)$$

so that the total number of customers in the system at time t is $N_{C_f}^t + Y_{C_f}^t$.

Since all queues in C_f are blowing up, there will be a time T^* such that for $t \geq T^*$ there are always customers to serve in each of the queues of C_f. Thus for any non-idling service policy the rate of flow of customers across C_f, f_{C_f}, can be calculated from equation (2). Note that $f_{C_f} \geq f^*$. Now fort $\geq T^*$ we *have:*

$$\begin{aligned} N_{C_f}^t + Y_{C_f}^t &= N_{C_f}^{T^*} + (A^t - A^{T^*}) - f_{C_f}(t - T^*) + Y_{C_f}^t \\ &\leq N_{C_f}^{T^*} + (A^t - A^{*}) - f^*(t - T^*) + Y_{C_f}^t \end{aligned}$$

taking the limit and dividing by t gives:

$$\begin{aligned} &\lim_{t\to\infty} \frac{N_{C_f}^t + Y^t}{t} \\ &= \lim_{t\to\infty} \frac{N_{C_f}^{T^*} + (A^t - A^{T^*}) - f_{C_f}(t - T^*) + Y_{C_f}^t}{t} \\ &\leq A - f^* \end{aligned}$$

This proves lemma 3.2

Proof of Theorem 3.1: This follows immediately from Lemmas 3.1 and 3.2. ■

4 Implementation Issues

here we discuss implementation issues associated with the routing/service policy. We first address the problem of choosing which queue to serve.

The service policy employed here is to serve any nonempty queue. Thus each server must poll all of its input queues during every slot to determine which have customers. We conjecture that the following service policy has tile same stability region under suitable assumptions. Service policy β: Assume that at some point in the past, the server knew the lengths of all of its input queues. Now, included in the header of each customer is the number of jobs that have arrived to the queue since the last customer was served. The

server then serves the longest connected queue *based on this information*. Thus the server will continue to serve the current queue until its length drops below the last reported length of some other connected queue, at which time the server will switch its attention.

Similarly, we conjecture that systems stability is robust with respect to delays in the queue length information. Periodically updating the path lengths to the destination could be done as with the queue lengths in the service policy β, with path length information attached to acknowledgements of received packets. This would greatly decrease the overhead associated with the LPP routing policy.

5 Conclusion

We have shown that for acyclic networks with random connectivities and routing, the policy that maximizes throughput is the one that routes customers along the shortest path to the destination at each node and serves the longest connected queue at each node. We are currently working on the proofs that the service policy β outlined above and its corresponding routing policy stabilize the system, as well as extending our existing proofs to the stationary and ergodic case.

References

[1] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, 39:466–478, 1993.

[2] Nicholas Bambos and George Michailidis. On the stationary dynamics of queueing networks with random topologies. In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 4, pages 3638-3613, 1995.

[3] Nicholas Bambos and George Michailidis. Minimizing the loss flow in parallel queues with finite buffers under random server connectivities and routing. *In Preparation.*

[4] Jean Walrand. *An Introduction to Queueing Networks*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.